**Digital Nurture 4.0 – Week 5**

Microservices with Spring Boot 3 and Spring Cloud

HANDSON EXERCISES

Filename :2 .Microservices with API gateway

**Exercise 1: Creating Microservices for account and loan**

**PART 1: Folder Setup**

1. Create Folder  
   In D: drive, create a new folder named with your employee ID, for example:  
   D:\1234567
2. Create 'microservices' Folder  
   Inside your employee ID folder, create a folder named:  
   D:\1234567\microservices

**PART 2: Account Microservice Setup**

1. Open <https://start.spring.io/>
2. Fill the form:
   * **Group:** com.cognizant
   * **Artifact:** account
   * **Dependencies:**
     + **Spring Boot DevTools**
     + **Spring Web**
3. Click **"Generate"** – it will download a zip.
4. Extract the zip, and **move** the extracted folder account into:

D:\1234567\microservices

1. Open **Command Prompt**, navigate into the account folder:

cd D:\1234567\microservices\account

1. Build the project using Maven:

mvn clean package

1. Open **Eclipse**, then:

* Go to **File > Import > Existing Maven Project**
* Select D:\1234567\microservices\account
* Finish

1. Inside the account project, create a **REST Controller**:  
   AccountController.java inside src/main/java/com/cognizant/account/controller

package com.cognizant.account.controller;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.RestController;

import java.util.Map;

public class AccountController {

public Map<String, Object> getAccountDetails(@PathVariable String number) {

return Map.of(

"number", number,

"type", "savings",

"balance", 234343

);

}

}

1. Run the main Spring Boot app (AccountApplication.java), and test in browser:

<http://localhost:8080/accounts/00987987973432>

**PART 3: Loan Microservice Setup**

1. Go back to <https://start.spring.io/>
2. Fill the form:
   * **Group:** com.cognizant
   * **Artifact:** loan
   * **Dependencies:**
     + **Spring Boot DevTools**
     + **Spring Web**
3. Click **"Generate"** and extract the zip
4. Move the loan folder into:

D:\1234567\microservices

1. Open **Command Prompt** and build:

cd D:\1234567\microservices\loan

mvn clean package

1. Import the loan project into Eclipse as Maven project.
2. Inside loan project, create controller:  
   LoanController.java inside src/main/java/com/cognizant/loan/controller

package com.cognizant.loan.controller;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.RestController;

import java.util.Map;

public class LoanController {

public Map<String, Object> getLoanDetails(@PathVariable String number) {

return Map.of(

"number", number,

"type", "car",

"loan", 400000,

"emi", 3258,

"tenure", 18

);

}

}

1. **Set different port** for Loan service:

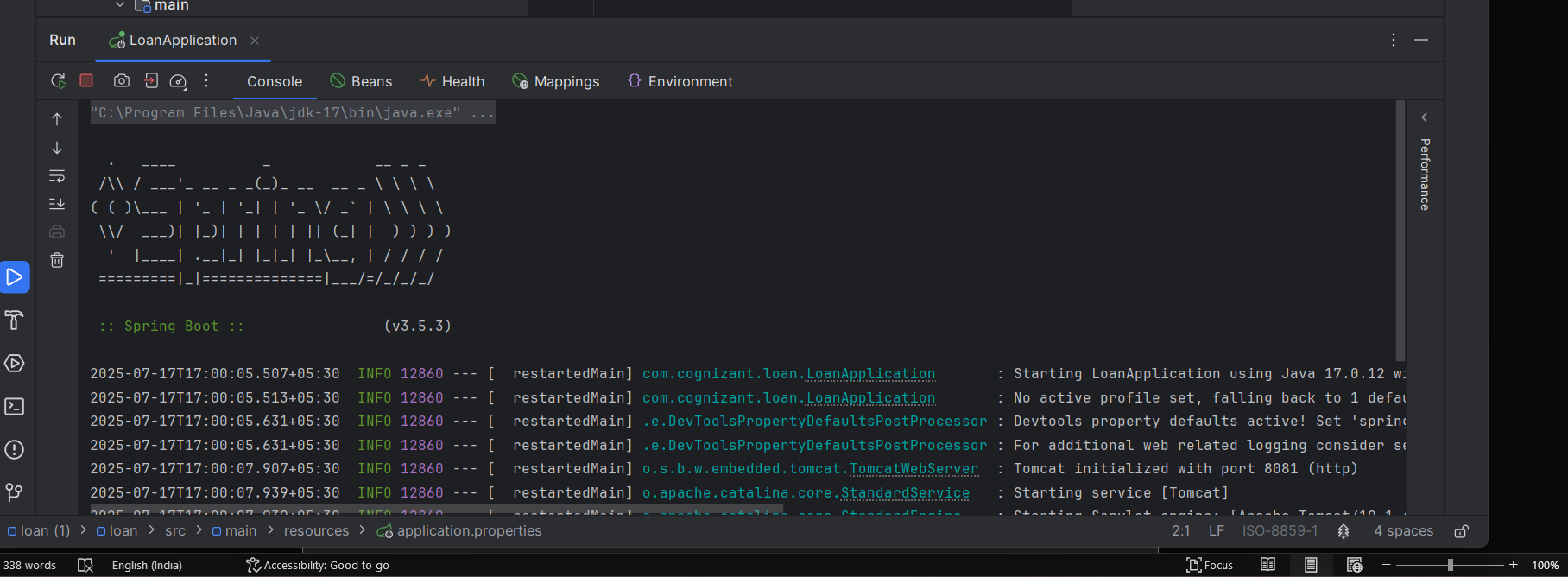
* Open src/main/resources/application.properties and add:

server.port=8081

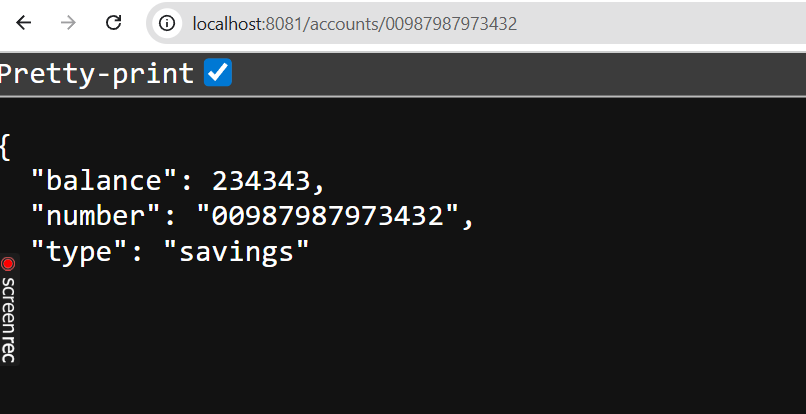
1. Run the LoanApplication.java Spring Boot app.
2. Test in browser:

<http://localhost:8081/loans/H00987987972342>

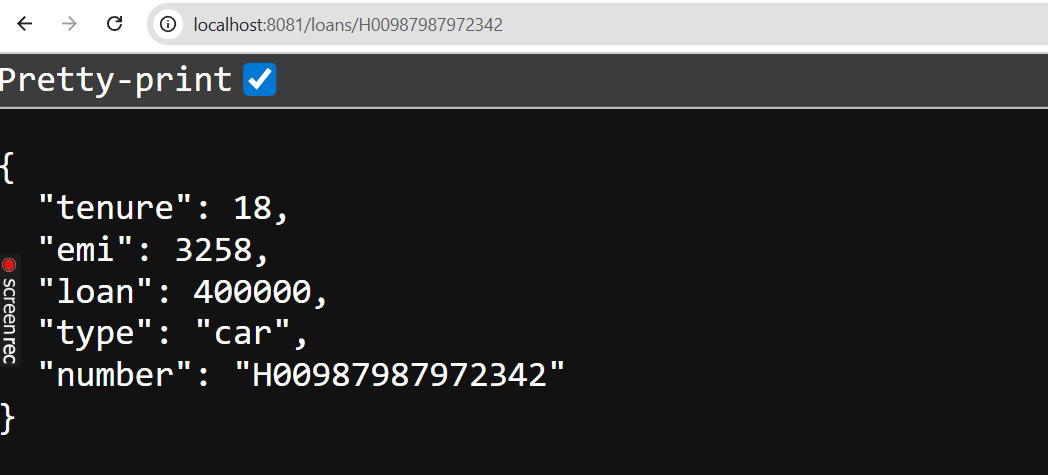
**OUTPUT:**



Output for account:



Output for loan:



**ADDITIONAL EXERCISES**

**Exercise 1:Create Eureka Discovery Server and register microservices**

Step 1: Create Eureka Server

Project Details:

* Project: Maven
* Language: Java
* Spring Boot: 3.2.4
* Group: com.cognizant
* Artifact: eurekaserver
* Name: eurekaserver
* Package Name: com.cognizant.eurekaserver
* Packaging: Jar
* Java version: 17
* Dependencies:
  + Spring Boot DevTools
  + Eureka Server

pom.xml:

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>3.2.4</version>

</parent>

<groupId>com.cognizant</groupId>

<artifactId>eurekaserver</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>eurekaserver</name>

<description>Eureka Server for service registry</description>

<properties>

<java.version>17</java.version>

<spring-cloud.version>2023.0.1</spring-cloud.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

</dependencies>

<dependencyManagement>

<dependencies>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-dependencies</artifactId>

<version>${spring-cloud.version}</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

**application.properties**

server.port=8761

spring.application.name=EUREKA-SERVER

eureka.client.register-with-eureka=false

eureka.client.fetch-registry=false

**EurekaServerApplication.java**

package com.cognizant.eurekaserver;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;

@EnableEurekaServer

public class EurekaServerApplication {

public static void main(String[] args) {

SpringApplication.run(EurekaServerApplication.class, args);

}

}

**Step 2: Run Eureka Server**

* Run EurekaServerApplication.java
* Access: http://localhost:8761/

**Step 3: Create Account Microservice**

**Project Details:**

* Group: com.cognizant
* Artifact: account
* Name: account
* Package: com.cognizant.account
* Port: 8081
* Dependencies:
  + Spring Boot DevTools
  + Spring Web
  + Eureka Discovery Client

**AccountController.java**

package com.cognizant.account.controller;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import java.util.Map;

@RestController

@RequestMapping("/account")

public class AccountController {

@GetMapping("/details")

public Map<String, Object> getAccountDetails() {

return Map.of(

"accountNumber", "1234567890",

"accountHolder", "Jane Doe",

"balance", 5000.75,

"status", "ACTIVE"

);

}

}

**AccountApplication.java:**

package com.cognizant.account;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.netflix.eureka.EnableEurekaClient;

@SpringBootApplication

@EnableEurekaClient

public class AccountApplication {

public static void main(String[] args) {

SpringApplication.run(AccountApplication.class, args);

}

}

**application.properties:**

server.port=8081

spring.application.name=ACCOUNT-SERVICE

eureka.client.service-url.defaultZone=http://localhost:8761/eureka/

eureka.instance.prefer-ip-address=true

management.endpoints.web.exposure.include=\*

management.endpoint.health.show-details=always

**Step 4: Create Loan Microservice**

**Project Details:**

* Group: com.cognizant
* Artifact: loan
* Name: loan
* Package: com.cognizant.loan
* Port: 8082
* Dependencies:
  + Spring Boot DevTools
  + Spring Web
  + Eureka Discovery Client

**LoanController.java**

package com.cognizant.loan.controller;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import java.util.Map;

@RestController

@RequestMapping("/loan")

public class LoanController {

@GetMapping("/details")

public Map<String, Object> getLoanDetails() {

return Map.of(

"loanId", "LN12345678",

"borrowerName", "John Smith",

"amount", 500000.0,

"interestRate", 7.5,

"status", "APPROVED"

);

}

}

LoanApplication.java:

package com.cognizant.loan;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.netflix.eureka.EnableEurekaClient;

@SpringBootApplication

@EnableEurekaClient

public class LoanApplication {

public static void main(String[] args) {

SpringApplication.run(LoanApplication.class, args);

}

}

**application.properties:**

server.port=8082

spring.application.name=LOAN-SERVICE

eureka.client.service-url.defaultZone=http://localhost:8761/eureka/

eureka.instance.prefer-ip-address=true

management.endpoints.web.exposure.include=\*

management.endpoint.health.show-details=always

**Step 5: Run All Applications**

* Start EurekaServerApplication (port 8761)
* Start AccountApplication (port 8081)
* Start LoanApplication (port 8082)
* Visit http://localhost:8761 to verify registered services.
* Test endpoints:
  + http://localhost:8081/account/details
  + <http://localhost:8082/loan/details>

**OUTPUT:**

